

# ALGORITMO GENÉTICO PARA UN PROBLEMA DE RUTEO CON ENTREGA Y RECOLECCIÓN DE PRODUCTO Y CON RESTRICCIONES DE VENTANA DE HORARIO

JESÚS FABIÁN LÓPEZ PÉREZ\*, MOHAMMAD H. BADI\*\*



El problema de entrega y recolección de producto con restricciones de ventanas de horario (Pickup and Delivery Problem with Time Window, PDP-TW)

puede verse en dos variantes principales. Savelsbergh, en 1995, distingue la variante para un solo vehículo SPDP-TW y la de vehículos múltiples MPDP-TW.<sup>1</sup> El primer caso se trata de un TSP-TW restrictivo (traveling salesman problem), mientras que el segundo se trata de un VRP-TW restrictivo (vehicle routing problem). En 1984 Solomon desarrolló 87 instancias de prueba para el VRP-TW, la más grande es de 100 clientes.<sup>2</sup> Hasta 1999 había 17 instancias que aún permanecían sin poder resolverse. Cook y Rich, de la Universidad de Rice en Houston, lograron resolver diez instancias en dicho año.<sup>3</sup> Nuestro problema se enfoca en el primer caso (SPDP-TW). La familia de problemas del tipo PDP-TW es más difícil de resolver que la del VRP-TW. La razón de lo anterior se debe a que el primero es una generalización del segundo.<sup>4</sup> El problema será planteado en: (1) objetivo y (2) condiciones de operación. La modelación matemática puede ser revisada en el anexo A.

## Objetivo del problema

El objetivo es determinar la ruta óptima para un vehículo de distribución. Una ruta se define como la secuencia de llegada a cada uno de los clientes, saliendo de un centro de distribución y regresando

a éste al final de la ruta. Se define como ruta óptima aquella que logre visitar todos los clientes de tal forma que el costo (o distancia o tiempo) incurrido sea el mínimo posible.

## Condiciones de operación para el problema

- El vehículo tiene una capacidad finita de carga.
- El vehículo, saliendo del centro de distribución, debe atender a un grupo de clientes geográficamente dispersos, y luego regresar al punto de origen.
- Se tiene una matriz de costo que define el tiempo o distancia requerida para ir de un cliente a otro.
- Cada cliente tiene un requerimiento de volumen de producto a entregar y un volumen a recoger.
- La cantidad de tiempo requerido para realizar la entrega y la recolección en cada cliente se considera la misma por tratarse de cargas de trabajo similares.
- La ventana de horario identificada para cada cliente está definida por una hora de apertura y una hora de cierre, la cual puede tener diferente amplitud, dependiendo de las características de cada cliente.

\* Facultad de Contaduría Pública y Administración, UANL.  
E-mail: fabian.lopez@e-arca.com.mx

\*\* Facultad de Ciencias Biológicas, UANL.

- g) La amplitud de la ventana de horario en cada cliente es igual a la diferencia entre la hora de cierre y la hora de apertura para la atención de cada cliente.
- h) La ventana de horario para un cliente puede ser modificada únicamente cuando, en el recorrido de la ruta óptima, existan otros clientes geográficamente cercanos y cuyas ventanas de horario sean diferentes.
- i) La visita a cada cliente debe ser aplicada dentro de la ventana de horario. No es permitido llegar antes de la hora de apertura ni después de la hora de cierre.
- j) De acuerdo a la secuencia de ruteo, se tendrá una descarga de producto a entregar en el cliente y una carga a recoger del cliente. En todo momento deberá de ser respetada la capacidad de carga del vehículo.

## Revisión bibliográfica de investigaciones previas

A continuación se expone un resumen bibliográfico de las investigaciones previas desarrolladas:

El punto de partida será el trabajo presentado por Ascheuer, Fischetti y Grotschel, en 2001, para el problema del TSP-TW, el cual es relativamente cercano en definición al SPDP-TW. Ascheuer, Fischetti y Grotschel mencionan en su trabajo que es interesante la falta de investigación en los métodos de ramificación y corte (Branch & Cut, BC) para la solución tanto de la versión simétrica (TSP-TW) como la asimétrica del problema (ATSP-TW). Como antecedente en este rubro mencionan a Baker, que en la

década de los 80 resolvió mediante algoritmos de BC instancias de hasta 50 nodos con ventanas de horario moderadamente cerradas. Matemáticamente, entre más cerradas sean las ventanas de horario más pequeño es el espacio solución a ser explorado al momento de resolver computacionalmente el problema. Las instancias que han sido típicamente experimentadas se caracterizan por ventanas de horario con un bajo porcentaje de traslape.<sup>9</sup>

Típicamente, la dificultad computacional para la solución de problemas de redes ha sido medida en términos de su tamaño (cantidad de nodos). Sin embargo, la dificultad del problema SPDP-TW depende fundamentalmente de la estructura de las ventanas de horario que se definan. Los resultados experimentales con el TSP-TW hechos por Ascheuer, Fischetti y Grotschel arrojan que este problema es particularmente difícil de resolver para instancias en las que la cantidad de nodos activos que contengan restricciones de ventana de horario estén por arriba del 50%. En lo referente al tamaño de las instancias experimentadas, sólo una de las heurísticas revisadas pudo resolver instancias de más de 69 nodos.<sup>10</sup>

Ascheuer, Fischetti y Grotschel experimentan con un método de solución especializado, con la finalidad de obtener mejores soluciones. Se experimentó con instancias de hasta 233 nodos. Para una instancia de 69 nodos se requirió de 5.95 minutos de tiempo computacional. Todas las instancias mayores ocuparon más de cinco horas de solución computacional, excepto una de 127 nodos.<sup>9</sup> Ascheuer, Fischetti y Grotschel, sobre la base de las experiencias computacionales, concluyen que las instancias del ATSP-TW, en el límite de 50 a 70 nodos, pueden ser resueltas hasta la optimalidad por el BC.

## Metodología propuesta

La propuesta de solución está compuesta por cuatro rutinas de preprocesamiento y un posprocesamiento con relación a la ejecución de la rutina del algoritmo genético. A continuación se enumeran las seis fases:

1. Fase de descomposición topológica de la red basado en un algoritmo SPP (Shortest path problem en inglés): las esquinas topológicas requeridas para la modelación de las restricciones de tránsito en la red son atendidas en una fase de preproceso, lo cual contribuye a disminuir la carga computacional durante la fase

Tabla I. Estudios de investigación previos.<sup>5-8</sup>

Investigador	Año	Variantes del Problema	Algoritmo	Dimensión	Variantes de Horario	Desempeño Computacional
Baker	1985	ATSP-TW	Ramificación y Corte (BC)	≤ 50 nodos	Cerradas, Poco Traslape	
Dumas, Desrosiers y Soufflot	1993	PDP	Flujo de Redes			
Bruggen, Lenstra y Schuur	1993	PDP-TW	Heurística Lin- Kernighan y Refinamiento Simulado	≤ 50 nodos	Cerradas, Poco Traslape	Instancias fuera de estas condiciones requieren ≥ 18 horas
Applegate y Goetz	1994	TSP-TW	Ramificación y Corte (BC)			
Van Elst	1995	VSP	Ramificación y Corte (BC)	≤ 15 nodos	Abiertas	
Coff y Vigo	1995	PDP-TW	Méto-Horizonte Tabu Search	58 ≤ n ≤ 130 nodos		Instancias para n ≥ 100 nodos requieren ≥ 1 Hora
Dumas y Salomon	1995	VSP-TW	Método de Descomposición de Costo y Corte	130 ≤ n ≤ 280 nodos	Cerradas, Poco Traslape	Instancias para 180 ≤ n ≤ 200 20 ≤ m ≤ 30
Salas y Gonsalves	1996	TSP-TW	Programación Dinámica	40 ≤ n ≤ 58 nodos	Cerradas, Poco Traslape	≤ 2 Horas
Mingozzi, Bianco y Fischetti	1997	TSP-TW	Híbrido BC y Programación Dinámica	≤ 128 nodos		
Ascheuer, Jørgen y Reinelt	2000	ATSP-TW	BC con generación de Membranas	50 ≤ n ≤ 78 nodos	Abiertas y con hasta 4% de incidencia de TW	90 ≤ m ≤ 20
Ascheuer, Fischetti y Grottschel	2001	ATSP-TW	Híbrido BC con Heurísticas de Intercambio	78 ≤ n ≤ 233 nodos	Cerradas, Poco Traslape	5 min a Tiempo < 8 hrs

posterior de optimización. De lo anterior, si partimos de un valor de  $N_1$  nodos en una red, obtendríamos un valor  $N_2$ , donde  $N_1 \approx 4N_2$ .

2. Fase de compresión/clusterización por maximización de afinidad geográfica (vecindad): los  $N_2$  nodos de la red se agrupan para obtener una cantidad reducida de  $N_3$  meta-nodos (donde:  $N_3 < N_2$ ). Las estructuras de las ventanas de horario asociadas a los  $N_2$  nodos requieren compatibilidad entre sí para poderse agrupar. En el algoritmo se utilizó un factor de compresión del 50% para la agrupación.
3. Fase de compresión/discriminante por heurística de los " $k$ " nodos más cercanos: se eliminan los arcos de la red que tengan mayor costo y, por tanto, menor probabilidad de aparecer en la solución óptima. Para cada nodo  $N_3$  se mantienen en la red sólo los " $k$ " arcos de menor costo, donde  $k \ll N_3$ . Se utilizó un factor discriminante del 20% para evitar dejar fuera del espacio de búsqueda la solución óptima.
4. Fase de generación agresiva de cortes: partiendo de los  $N_3$  meta-nodos, el objetivo es encontrar, lo más rápidamente posible, una solución factible que cubra las restricciones de ventana de horario y de carga del vehículo. La lógica para generar los cortes identifica el nodo del *tour* que tiene la mayor desviación respecto a la ventana de horario y la capacidad de carga del vehículo (nodo pivote). Luego se verifican los nodos del *tour* que pueden ser identificados como afines, a efecto de re-secuenciar la posición del nodo pivote en el *tour*. El corte asegura que el nodo pivote " $k$ " utilice al menos uno de los arcos que lo conecten a uno de los nodos afines " $j$ ".

$$\begin{aligned}
 &\text{sea } I = \{1..N_3\} \text{ (nodos de la red)} \\
 &k \in I \text{ (nodo pivote)} \\
 &j \in I \text{ (nodos afines a } k) \\
 &\sum_{i=1}^m (x_{ik} + x_{kj}) \geq 1 \quad \forall k \in I
 \end{aligned}$$

Los cortes generados se guardan en un *pool* de restricciones. El procedimiento continúa hasta que se encuentra la primera solución factible al problema.

5. Fase evolutiva: el objetivo es aproximar la solución óptima para la versión compacta de la red. Mantener en el *pool* de restricciones un corte

innecesariamente equivale a estar en riesgo de dejar fuera del espacio de búsqueda mejores soluciones que la que actualmente se tiene. La experiencia computacional indica que la cantidad de cortes que llegan a acumularse en el "*pool*" de restricciones es significativo (15-40 cortes). El objetivo es identificar cuáles cortes del "*pool*" es necesario eliminar. La eliminación de cortes no puede ser vista de forma individual para cada uno, ya que la presencia y la eliminación de un corte puede comprometer simultáneamente la presencia y la eliminación de otro(s). Identificar cuáles cortes conviene eliminar debe ser visto como un subproblema combinatorio, el cual es apropiado para atenderse mediante estrategias de recombinación evolutiva. Una codificación binaria permite representar la eliminación (0) y la presencia (1) del corte en el *pool*. El algoritmo genético aplica operadores de selección tipo torneo con un factor de cruzamiento al 50%. El método de reproducción se aplica a través de dos puntos de cruce aleatorios a lo largo de la longitud cromosómica. El factor de mutación es inicializado al 5% y autoajustado en cada generación de acuerdo al porcentaje de individuos en la población que sean idénticos genéticamente hablando. Al aumentar el nivel de degeneración en la población, se aplica una curva exponencial de crecimiento en el porcentaje de mutación con un límite asintótico al 50%. El factor de elitismo se limita a un 15% de la población.

6. Fase de descompresión para desagregar la ruta propuesta para la red original: el posprocesamiento tiene el objetivo de traducir la solución obtenida en la red compacta en una que sea topológicamente equivalente a la red original. La primera rutina de desagregación está enfocada a determinar la secuencia óptima en la cual los  $N_3$  meta-nodos se deben desagregar para volver a formar los  $N_2$  nodos obtenidos en la fase 2. Lo anterior se logra a través de un algoritmo de secuenciamiento de baja escala, lo cual matemáticamente puede ser resuelto hasta la optimalidad en un tiempo polinomial. La segunda rutina de reconversión emplea la información topológica generada en la fase 1. Su objetivo es sustituir la secuencia del *tour* definido por los  $N_2$  nodos de acuerdo a la cadena de movimientos cardinales que se requieren para obtener los  $N_1$  nodos de la red original.

## Objetivo e hipótesis del problema de investigación

El objetivo es comprobar si el método propuesto, basado en un algoritmo genético (AG), es viable para encontrar soluciones razonablemente buenas ( $\geq 90\%$  optimalidad) y a un bajo costo computacional ( $\leq 5$  minutos) para resolver el problema del SPDP-TW, resumiendo:

$$H_0: \mu_{\text{Grupo experimental}} \leq 90\% (\mu_{\text{Grupo Control}})$$

$$H_a: \mu_{\text{Grupo experimental}} > 90\% (\mu_{\text{Grupo Control}})$$

donde  $\mu$  es la media poblacional correspondiente a los porcentajes de optimalidad alcanzados en la medición del quinto minuto computacional para instancias de tamaño  $100 \leq w \leq 120$ , (donde  $w$  = cantidad de nodos en la red). La comprobación de la hipótesis se hará a través de una prueba de diferencias de medias de "T" de Student:

$$t = \frac{D - \mu_D}{\frac{S_D}{\sqrt{n}}}, \text{ donde: } D = \frac{\sum_{i=1}^n D_i}{n}, \text{ y } S_D = \sqrt{\frac{\sum_{i=1}^n (D_i - D)^2}{n-1}}$$

## Justificación científica y relevancia social

- El problema planteado es de naturaleza combinatoria y está catalogado como NP-Hard.<sup>11</sup>
- En relación al área de aplicación, la variante menos investigada del SPDP-TW es la que tiene que ver con la distribución física de producto.<sup>12</sup>
- Las instancias del problema de investigación encontradas en la práctica de las empresas son arriba de 70 clientes por ruta y con ventanas de horario anchas (anchura  $\geq 50\%$ ). Esta propiedad ocasiona que el problema tenga un espacio solución grande. Esta característica no se enfrenta en la investigación de Ascheuer, Fischetti y Grottschel.
- Los algoritmos de solución exacta (BC), para solucionar instancias como las antes descritas, requieren un tiempo computacional que excede los límites prácticos de la operación de una empresa.
- Existe la necesidad de desarrollar algoritmos que puedan ofrecer soluciones que aunque no sean matemáticamente óptimas, sí sean razonablemente de calidad ( $\geq 90\%$  del óptimo) y que pue-

dan obtenerse en un tiempo computacional práctico ( $\leq 5$  minutos).

- A partir de investigaciones de campo en empresas del área metropolitana de la ciudad de Monterrey (AMM).<sup>13</sup>
  - Más del 20% de las empresas enfrentan problemas relacionados con el SPDP-TW.
  - Más del 32% de las empresas resuelven el problema a través de la empírica del administrador.
  - Menos del 12% de las empresas del AMM ha utilizado tecnología APS (Advanced Planning Scheduling) para dar tratamiento al problema sin que exista evidencia de éxito.

## Variable para la prueba experimental

"El porcentaje de optimalidad relativa" será la variable a utilizar para la prueba estadística de la hipótesis. Esta métrica de calidad de la solución se mide con respecto a la solución registrada por el grupo control mediante un algoritmo de ramificación y corte (BC). Mientras en el AG se busca medir con qué rapidez se mejora una solución, con el algoritmo BC se obtiene el valor de referencia para medir el desempeño del AG. Con esto determinaremos la relación que guarda el tiempo de ejecución computacional versus el porcentaje de optimalidad.

## Metodología y desarrollo experimental

La técnica que se aplicará para la prueba de la hipótesis será a través de un "diseño de experimentos". El objetivo del diseño de experimentos es determinar si acaso al utilizar un determinado tratamiento, se produce alguna mejora o no en el proceso.<sup>14</sup> La medición de la variable "porcentaje de optimalidad relativa" se hará mediante la aplicación de cuatro instrumentos experimentales:

- Algoritmo de ramificación y corte (BC) como método de solución exacta (grupo control).
- Algoritmo genético básico 1: motor de optimización "Evolver" © Versión 6.0 de Palisade Corporation.
- Algoritmo genético básico 2: motor de optimización "Solver" © Versión 4.0 de Frontline System.
- Algoritmo genético propuesto 3.

Para los cuatro instrumentos descritos, la medición del “porcentaje de optimalidad” se aplicó en cuatro momentos sucesivos de tiempo durante la ejecución del experimento: al minuto tres, cinco, ocho y diez. El primer instrumento basado en el algoritmo BC estuvo sujeto a un límite de tiempo de cinco horas, ya que su objetivo fue obtener la referencia de optimalidad para cada unidad experimental. El diseño experimental se aplicó para una muestra de 40 instancias, que tuvieron una complejidad matemática lo suficientemente intensa como para consumir un mínimo de diez minutos de esfuerzo computacional para su solución exacta mediante el algoritmo BC.

La matriz experimental se expone en la tabla II, la cual está formada por 16 tratamientos (cuatro instrumentos algorítmicos y cuatro intervalos de tiempo). Sobre la base de 40 unidades experimentales, entonces estamos hablando de un total de 640 mediciones. A partir de las condiciones de normalidad que fueron procuradas al establecer el tamaño de la muestra, se calculan los parámetros estadísticos para la variable “porcentaje de optimalidad relativa”:

1. Media muestral ( $m$ ).
2. Desviación estándar muestral ( $s$ ).

Tabla II. Matriz resultante del diseño del experimento.

Estadístico ( $m$ , $s$ ) para la Variable “% de Optimalidad Relativa”					
Mediciones a ser aplicadas después de cierto tiempo computacional	Instrumentos de Medición a ser comparados				
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto
	3er Minuto	(m11, s11)	(m12, s12)	(m13, s13)	(m14, s14)
	5to Minuto	(m21, s21)	(m22, s22)	(m23, s23)	(m24, s24)
	8vo Minuto	(m31, s31)	(m32, s32)	(m33, s33)	(m34, s34)
	10mo Minuto	(m41, s41)	(m42, s42)	(m43, s43)	(m44, s44)

La prueba estadística utilizada para la comprobación de la hipótesis fue la prueba “T” de Student. La aplicación de la prueba “T” en cada estadístico ( $m_{ij}$ ,  $s_{ij}$ ) calculará la probabilidad de que el instrumento algorítmico “j” en el intervalo de tiempo “i” obtenga un “porcentaje de optimalidad” mayor al 90% “ $P(x > 90\%)$ ”.

## Resultados del experimento

Las instancias aplicadas en el experimento se desarrollaron a partir de redes arriba de 100 nodos ( $100 \leq w \leq 120$ ). El experimento dio tratamiento a unidades experimentales con ventanas de horario activas

por arriba del 70% y con una amplitud mínima del 75%. Podemos confirmar que las condiciones de complejidad matemática estuvieron plenamente cubiertas y, por tanto, la prueba aplicada al AG propuesto es significativa. Para la implementación de los AG’s 2, 3 y 4, los parámetros operativos fueron fijados empíricamente a un mismo valor para hacer comparables sus desempeños y así eliminar la existencia de cualquier fuente de variabilidad no planificada en el experimento. Las condiciones computacionales aplicadas para el experimento fueron:

- a) Sistema operativo “Windows XP ©”
  - b) Procesador computacional INTEL © a 2.4 Ghz.
  - c) Memoria de acceso inmediato de 128 Mhz.
- En la tabla III se muestran los valores calculados para el estadístico “T”.

Tabla III. Matriz de valores calculados para el estadístico T.

Resultados: Matriz de valores calculados para el estadístico T							
Instrumentos de Medición a ser comparados							
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto		
					P(x > 90%)	P(x > 92.5%)	P(x > 95%)
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA	-0.4039	-0.5581	2.4261	1.0688	-0.0911
	5to Minuto	-2.4260	-0.1159	-0.3068	3.3130	1.5392	0.1111
	8avo Minuto	-1.2800	0.1616	0.3173	4.8510	4.1050	0.8300
	10mo Minuto	-0.7001	0.4001	0.9030	6.2980	5.5770	1.3277

La comprobación de la hipótesis se tiene al comparar el valor calculado de “T” versus su valor crítico. Para un 95% de confiabilidad la referencia mínima es “1.685”. Para un 99% de confiabilidad la referencia mínima sería “2.426”. Para cada valor calculado de “T” es posible establecer su correspondiente nivel de significancia. En la tabla IV se muestran los coeficientes de probabilidad “P Valor” para el estadístico “T”, donde  $P(x > Opt\%)$ .

Tabla IV. Valores de probabilidad “P Valor”.

Matriz de Probabilidad "P Valor" para el Estadístico T, $P(x > Opt\%)$							
		Instrumentos de Medición a ser comparados					
		Algoritmo BB (Grupo Control)	Algoritmo Genético Básico (Evolver)	Algoritmo Evolutivo Básico (Frontline)	Algoritmo Propuesto		
					$P(x > 90\%)$	$P(x > 92.5\%)$	$P(x > 95\%)$
Mediciones a ser aplicadas después de cierto tiempo computacional	3er Minuto	NA	34%	29%	99%	85%	48%
	5to Minuto	<1%	45%	38%	100%	93%	54%
	8vo Minuto	10%	56%	62%	100%	100%	79%
	10mo Minuto	24%	65%	81%	100%	100%	90%



## Discusión de los resultados del experimento

- a) El algoritmo BC invariablemente obtiene si no la solución óptima, sí una mejor solución que los AG's.
  - b) Los tiempos computacionales para el BC oscilan entre 20 minutos y cinco horas (límite de tiempo), dependiendo de cada unidad experimental tratada.
  - c) Se probaron instancias sencillas (nodos  $\leq 70$ , nodos con ventanas activas  $\leq 60\%$ ) para revisar el desempeño del BC. El resultado fue favorable con tiempos computacionales abajo de los tres minutos.
  - d) El AG 3 propuesto obtiene soluciones satisfactorias ( $\leq$  del 90% de optimalidad) y en tiempos computacionales razonables ( $3 \leq t \leq 5$  minutos).
  - e) En los AG's básicos (1 y 2), los porcentajes de optimalidad resultan ser significativamente inferiores ya que estos nunca sobrepasan el 90%.
  - f) Aunque el AG 1 evoluciona mucho más rápido que el AG 2, el AG 2 ofrece mejores porcentajes de optimalidad que el AG 1.
  - g) Aunque en los primeros tres minutos de ejecución el AG 1 ofrece mejor desempeño que el AG 2, al avanzar el tiempo computacional el AG 1 se atasca en un óptimo local, degenerando prematuramente la población de organismos.
  - h) El AG 2 sigue mejorando la población y obtiene una mejor solución. La diferencia observada en el desempeño de estos tres AG's se explica como sigue:
    - El AG 2 obtiene mejores soluciones que el AG 1, debido a la propiedad generacional en su método de reproducción.<sup>15</sup>
    - El AG 3 (propuesto) obtiene mejores soluciones que los otros dos, debido a que aprovecha del AG 2 la propiedad generacional de reproducción y, además, debido a la explotación de la estructura del problema.
- (2001).
  - II. El algoritmo BC obtiene la solución óptima para 38 de las 40 instancias, que son particularmente difíciles de resolver y en las cuales la investigación se concentra.
  - III. La hipótesis de investigación se comprueba: podemos establecer a un 100% de confiabilidad que el AG propuesto alcanza un porcentaje de optimalidad  $\geq 90\%$  en un tiempo computacional  $\leq 5$  minutos.
  - IV. El AG propuesto ofrece soluciones al problema de investigación dentro de un rango de optimalidad aceptable y en tiempos de ejecución computacionales que hacen factible su implementación en la práctica:
    - El AG propuesto alcanza un porcentaje de optimalidad  $\geq 90\%$  en un tiempo computacional  $\leq 3$  minutos a un 99% de confiabilidad.
    - El AG propuesto alcanza un porcentaje de optimalidad  $\geq 92.5\%$  en un tiempo computacional  $\leq 5$  minutos a un 93% de confiabilidad.
    - El AG propuesto alcanza un porcentaje de optimalidad  $\geq 92.5\%$  en un tiempo computacional  $\leq 8$  minutos un 100% de confiabilidad.
  - V. Sin embargo, también debemos establecer que el AG propuesto sólo puede asegurar un 90% de confiabilidad para alcanzar soluciones  $\geq 95\%$  de optimalidad y en tiempos computacionales  $\leq 10$  minutos.
  - VI. La expectativa de solución de los algoritmos genéticos analizados se ve sensiblemente afectada en la medida en la cual se requieran soluciones que se acerquen al óptimo verdadero (exacto).
  - VII. El AG propuesto sólo ofrece un 54% de confiabilidad cuando se requiere alcanzar un porcentaje de optimalidad  $\geq 95\%$  en un tiempo computacional  $\leq 5$  minutos. Lo anterior se traduce en el experimento al poderse alcanzar una optimalidad igual o mayor del 95% sólo en 22 instancias de las 40 unidades experimentales que fueron aplicados para un límite de cinco minutos de tiempo computacional.

## Conclusiones

A partir de los resultados expuestos concluimos que:

- I. El algoritmo BC (grupo control) explota eficazmente la estructura matemática del problema, alcanzando la solución óptima para instancias comparables a las de Ascheuer, Fischetti y Grotschel
- II. El investigador prevé que el desempeño obtenido por ambas propuestas algorítmicas (BC y AG) puedan servir para generar nuevas líneas de investigación para atender:
  1. Instancias múltiples de los problemas de ruteo, tales como el VRP-TW y el MPDP-TW.
  2. Instancias "no estáticas" de los problemas de ruteo,

las cuales se encuentran frecuentemente en las empresas con ambientes de despacho dinámico.

## Resumen

En la solución cuantitativa de problemas combinatorios es importante evaluar el costo-beneficio que surge al obtener soluciones de alta calidad en detrimento de los recursos económicos y computacionales requeridos. Tomando en cuenta el criterio anterior, se desarrolla un diseño experimental y a la vez un análisis comparativo entre un algoritmo de solución exacta basado en técnicas de ramificación y corte versus un algoritmo genético especializado para aplicarse en un problema frecuentemente encontrado en el ámbito de la logística de distribución. Los resultados son favorables.

**Palabras clave:** Logística, Investigación de operaciones, Algoritmos genéticos, Ruteo con ventanas de horario, NP-Hard.

## Abstract

Our problem is about a routing of a vehicle with pickup and delivery of product with time window constraints. This problem requires to be attended with instances of medium scale ( $\text{nodes} \geq 100$ ). A strong active time window exists ( $\geq 90\%$ ) with a large factor of amplitude ( $\geq 75\%$ ). This problem is NP-hard and for such motive the application of an exact method is limited by the computational time. This paper proposes a specialized genetic algorithm. We report good solutions in computational times below 5 minutes. This feature allows its application in business where the time decision is critical.

**Keywords:** Logistics, Operations research, Genetic algorithms, NP-Hard, Time Windows.

## Referencias

1. Savelsberg, M. (1995). Local search in Routing Problem with Time Windows, *Annals of Operations Research*, Rotherdam Holanda. Pp. 1-7.
2. Solomon, M. (1984). On the worst-case performance of some heuristics for the vehicle routing and scheduling problem with time window constraints, "Report 83-05-03", The Wharton School, EUA.
3. Cook, W; Rich, Jennifer. (1999). A parallel cutting-plane algorithm for the vehicle routing problem with time windows, *Computational and Applied Mathematics* Rice University, Houston EUA. P. 5.
4. Palmgren, Myrna. (2001). A Column Generation Algorithm for the Log Truck Scheduling Problem, Department of Science and Technology (ITN), Linköping University, Norrköping Sweden. P. 3.
5. Applegate, D; Bixby, R; Chvátal, V. (1998). On the solution of traveling salesman problems. "Documenta Mathematica Extra Volume ICM III", EUA. Pp. 45-56.
6. Dumas, Y; Desrosiers, J; Solomon, M. (1995). An algorithm for the traveling salesman problem with time windows, "Operations Research 43(2)", EUA. Pp. 23-25.
7. Eijl Van, C. (1995). A polyhedral approach to the delivery man problem, "Technical Report 95-19", Department of Mathematics and Computer Science, Eindhoven University of Technology, The Neatherlands. Pp. 12-14.
8. Lenstra, K. (1990). A Variable Depth Approach for the Single-Vehicle Pickup and Delivery Problem with Time Windows, "COSOR No. 90-48", Eindhoven University of Technology, Alemania. Pp. 35-90.
9. Ascheuer, N; Fischetti, M; Grotschel, M. (2001). Solving ATSP with time windows by branch-and-cut, Springer-Verlag, Alemania.
10. Ascheuer, N; Jünger, M; Reinelt, G. (2000). A branch & cut algorithm for the asymmetric Traveling Salesman Problem with precedence constraints, "Computational Optimization and Applications 17(1)", EUA. Pp. 2-7.
11. Tsitsiklis, J. (1992). Special cases of traveling salesman and repairman problems with time windows, "Networks No. 22", EUA.
12. Mitrovic, Snezana. (1998). Pickup and Delivery Problem with Time Windows, "Technical Report SFU CMPT TR 1998-12", Canada. Pp. 38-39.
13. López, Fabián. Aplicación de un AG para un problema de logística de ruteo; Tesis Doctoral no publicada, Cap. 2-3.
14. Fisher, R. (1971). The Design of Experiments, Hafner Press & Macmillan Publishers, London. Pp. 25-50.
15. Goldberg, D. (1995). Genetic Algorithms in Search, Optimization and Machine Learning, Addison Wesley Pub Co, University of Massachusetts, EUA.